universidade anhembi morumbi
Mundialmente criativa e inovadora
Laureate International Universities®

Introdução a Programação para Games

Prof. Alexandre Nascimento Tomoyose

2º semestre 2016

universidade anhembi morumbi
Mundialmente criativa e inovadora
Laureate International Universities®

Expressões e Condicional

Prof. Alexandre Nascimento Tomoyose

2º semestre 2016

Expressões

- **Definição**
  - Uma expressão é a combinação de diversos valores e operadores que pode ser avaliada em um único valor resultante
  - A avaliação ou cálculo da expressão é realizada seguindo uma ordem de prioridade predefinida

Expressões

- Regras
  - Os operadores respeitam a seguinte ordem de prioridade
    - 1° - Unários
    - 2° - Aritméticos
    - 3° - Relacionais
    - 4° - Lógicos
  - Trechos entre parênteses tem sempre prioridade

Expressões

- Operadores Unários
  - Precisam de apenas um valor para realizar a operação
    - ++ (incrementa)
    - -- (decrementa)
    - ! (NOT)

Expressões

- Operadores Aritméticos
  - Realizam operações matemáticas, resultando em um número:
    - + (soma)
    - * (multiplicação)
    - - (subtração)
    - / (divisão)
    - % (resto da divisão)

Expressões

- Operadores Aritméticos
  - Seguem a mesma prioridade das operações na matemática
  - Exemplos
    - media = (a + b + c) / 3
    - media_ponderada = (2.5 * a + 4 * b + 3.5 * c) / 10

Expressões

- **Exercício**
  - 4 / 5
  - 5 / 4
  - 4 / 5.0
  - 5.0 / 4
  - 4.0 + 5 % 4
  - (4.0 + 5) % 4
  - 5 + 4 * 5 + 4 / 4 * 5
  - (5 + 4) * 5 + 4 / (4 * 5)

Expressões

- Operadores Relacionais
  - Comparam dois valores, resultando em verdadeiro ou falso:
    - == (igual)
    - != (diferente)
    - > (maior)
    - < (menor)
    - >= (maior ou igual)
    - <= (menor ou igual)

Expressões

- Operadores Relacionais
  - Os operadores igual e diferente tem menor prioridade em relação aos outros, ou seja, são resolvidos depois
  - Exemplos
    - valor = 2 < 1
    - valor2 = false != true

Expressões

- **Exercício**
  - 4 < 4
  - 5 == 5
  - 1 + 2 >= 3
  - 5 != 3 + 2
  - true != 2 < 1
  - false == 4 < 5 != true
  - 2 > 3 != 3 < 2 != true
  - false != true == false != true

Expressões

- **Operadores Lógicos**
  - **Realizam operações entre booleanos, que permitem apenas os valores verdadeiro ou falso:**
    - **&& (AND)**
    - **|| (OR)**
    - **! (NOT)**

Expressões

- Operadores Lógicos
  - O operador ! (NOT) tem prioridade por se tratar de um operador unário (requer apenas um valor)
  - Exemplos
    - teste = !true
    - teste1 = false && true
    - teste2 = false || true

Expressões

- **Exercício**
  - 1 != 2 && 3 == 3
  - 1 == 2 && 2 != 3
  - 1 < 2 || 3 == 2
  - 2 < 1 || 3 < 4 || 6 < 5
  - 2 < 1 && 3 < 4 || 6 < 5
  - 2 < 1 || 3 < 4 && 6 < 5

Expressões

- **Tabela da Verdade**

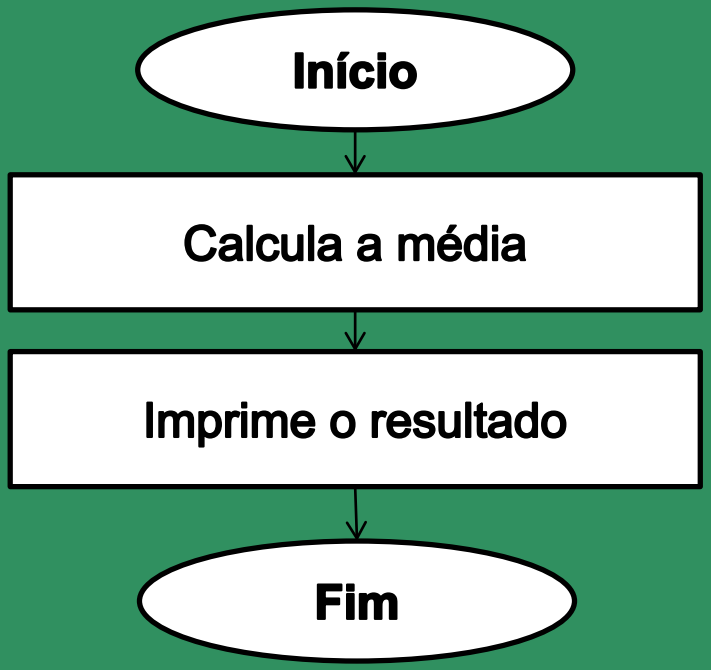
A	B	A || B	A && B	!A
F	F	F	F	V
F	V	V	F	V
V	F	V	F	F
V	V	V	V	F

Estruturas de Controle

- Qualquer programa pode ser escrito utilizando-se apenas três estruturas:
  - Estrutura de Sequência
  - Estrutura de Seleção (Condicional)
  - Estrutura de Repetição (Laço)

Estruturas de Controle

- **Estrutura de Sequência**
  - As instruções são executadas uma após a outra, respeitando-se a ordem

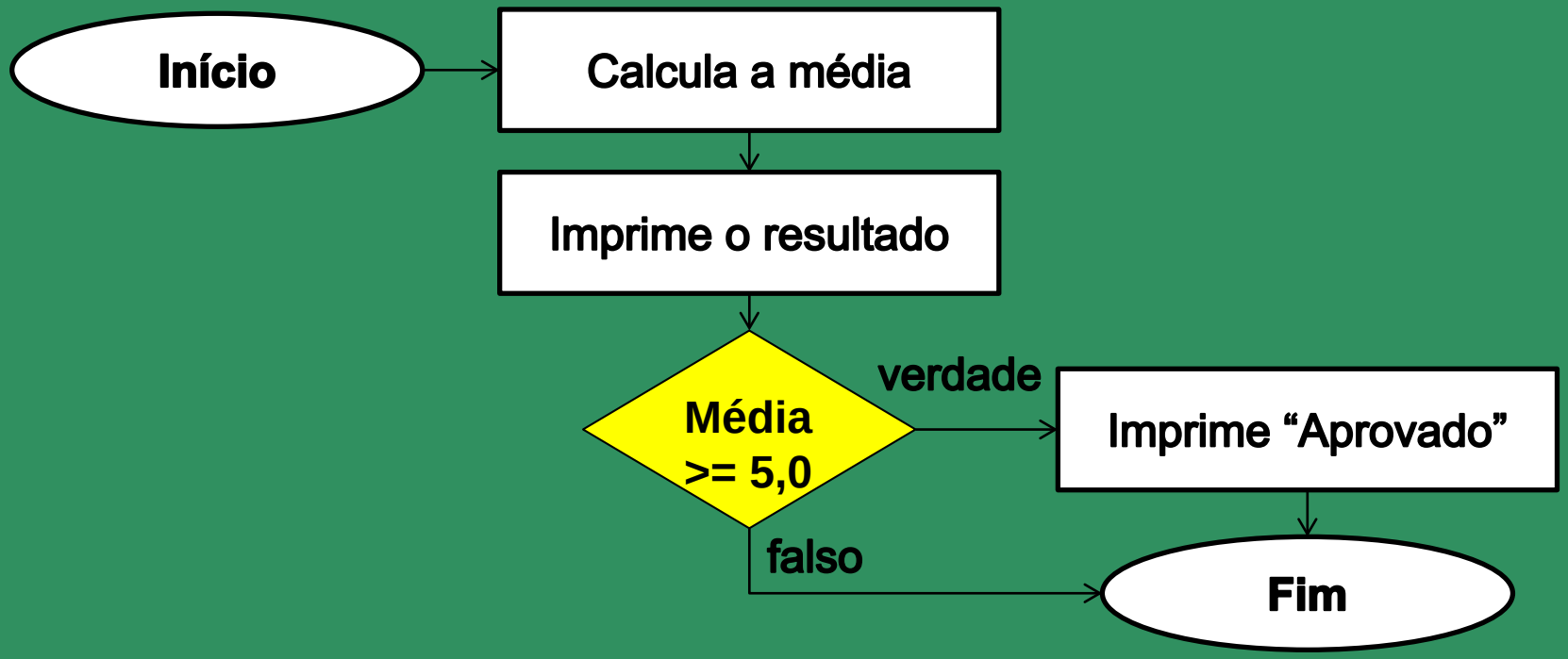


Estruturas de Controle

- **Exercício**
  - **Escreva um algoritmo que calcula a média a partir de 3 notas e imprime o resultado.**
  - **Além disso, caso a nota seja 5,0 ou maior imprima "Aprovado".**

Estruturas de Controle

- **Estrutura de Seleção (Condicional)**
  - Escolhe um curso de ação, dependendo de uma determinada condição



Estruturas de Controle

- **Estrutura de Seleção (Condicional)**
  - O comando **if** é utilizado para realizar uma condicional
  - Esta instrução permite que um trecho de código seja executado apenas quando uma determinada condição for verdadeira

Estruturas de Controle

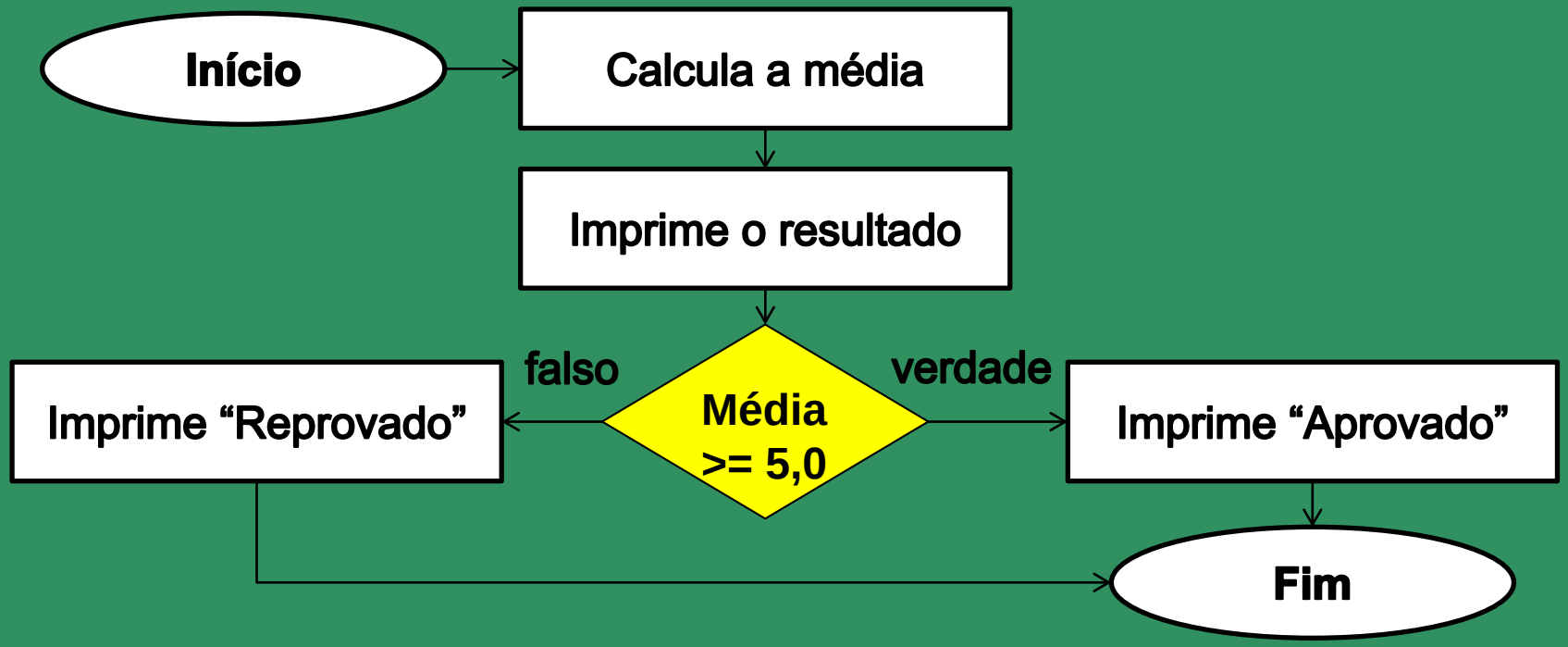
- **Estrutura de Seleção (Condicional)**
  - A sintaxe desta instrução segue o padrão:
    - if (condição) { comandos }
  - A **condição** pode ser uma variável booleana ou qualquer expressão que resulte em verdadeiro ou falso
  - Os **comandos** serão executados apenas quando a condição for verdadeira

Estruturas de Controle

- **Exercício**
  - **Escreva um algoritmo que dado um número inteiro imprime se este é par ou ímpar.**

Estruturas de Controle

- **Estrutura de Seleção (Condicional)**
  - Também é possível estabelecer uma ação no caso da condição ser falsa



Estruturas de Controle

- **Estrutura de Seleção (Condicional)**
  - O comando **if** permite a definição de comandos a serem executados caso a condição seja falsa através da instrução **else**
  - A sintaxe completa desta instrução segue o padrão:
    - if (condição) { comandos }
    - else { comandos2 }

Estruturas de Controle

- **Exercício**
  - Escreva um algoritmo que dados três valores A, B e C, verifique se estes podem ser os comprimentos dos lados de um triângulo.
  - Caso positivo, escrever uma mensagem informando qual tipo de triângulo: equilátero, isósceles ou escaleno.
  - Caso negativo, escreva “Não é triângulo”.

Estruturas de Controle

- **Exercício**
  - O comprimento de um lado do triângulo é sempre menor do que a soma dos outros dois.
  - A classificação segue a regra:
    - Equilátero - Todos os lados iguais
    - Isósceles - Dois lados iguais
    - Escaleno - Todos os lados diferentes

Estruturas de Controle

- **Exercício**
  - Escreva um algoritmo que dado um mês e um ano imprime o número de dias daquele mês.
  - Exemplos:
    - mes = “janeiro” / ano = 2016
    - mes = “novembro” / ano = 1985
    - mes = “fevereiro” / ano = 2000
  - Considere os anos bissextos!

universidade anhembi morumbi
Mundialmente criativa e inovadora
Laureate International Universities®

Processing

Prof. Alexandre Nascimento Tomoyose

2º semestre 2016

Processing

- **Exemplo**
  - *//* Determina o tamanho da fonte
  - textSize(60);
  - *//* Escreve na tela
  - text(“texto”, 200, 300);
  - *//* O comando fill determina a cor do texto

Processing

- **Exercício**
  - Faça o algoritmo que determina se um número é par ou ímpar
  - Escreva o texto em azul se for par
  - Escreva o texto em vermelho se for ímpar